

Figure 1

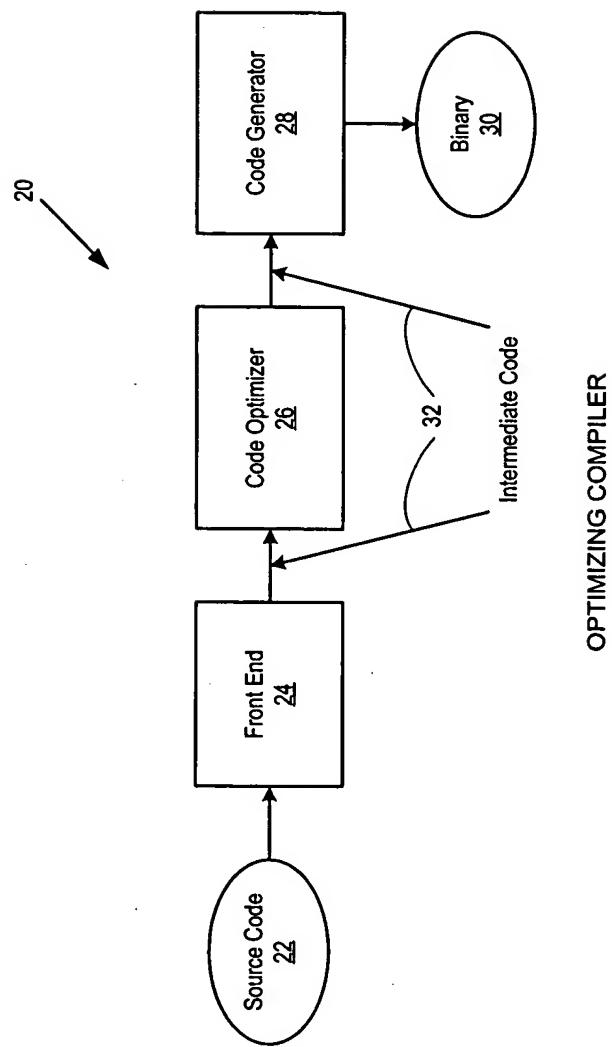


Figure 2

OPTIMIZING COMPILER

Fig 3. Compilation Process of Profile-Directed Optimizations

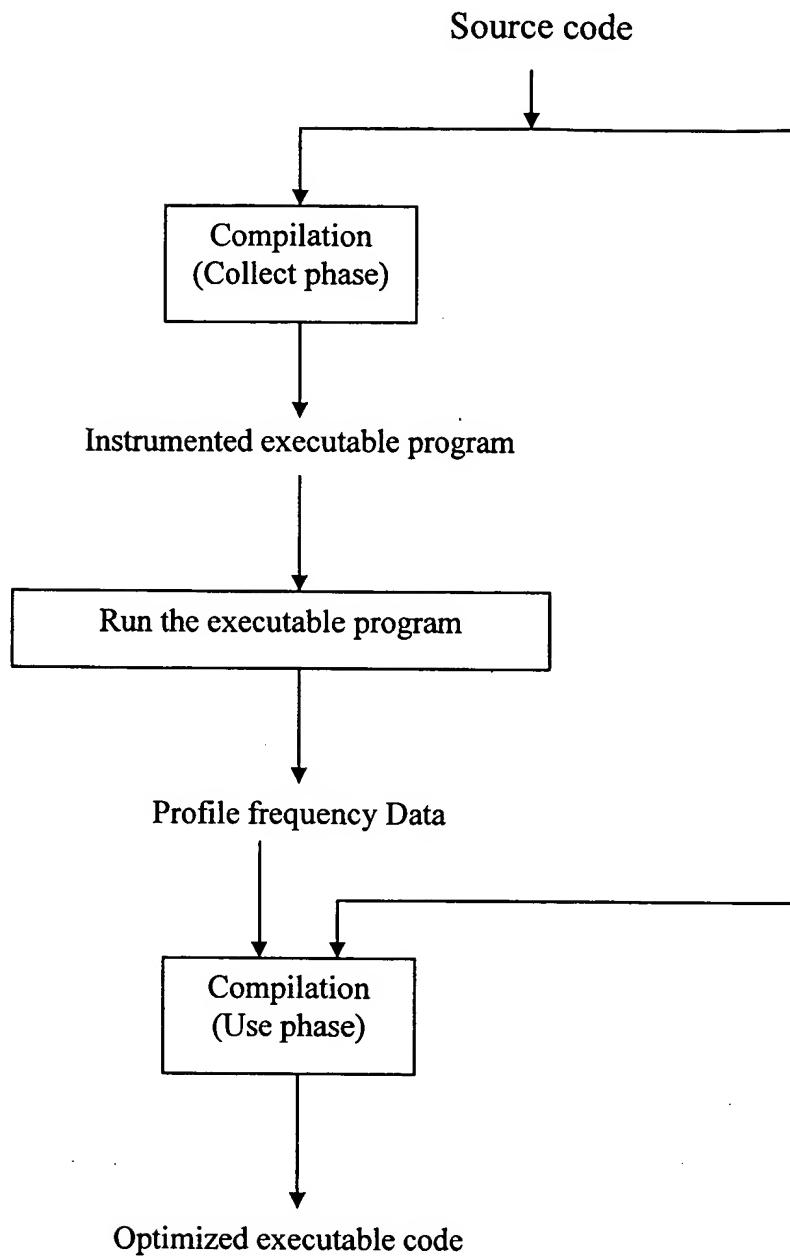


Fig 4. Compiler Components: collect phase and use phase (prior art)

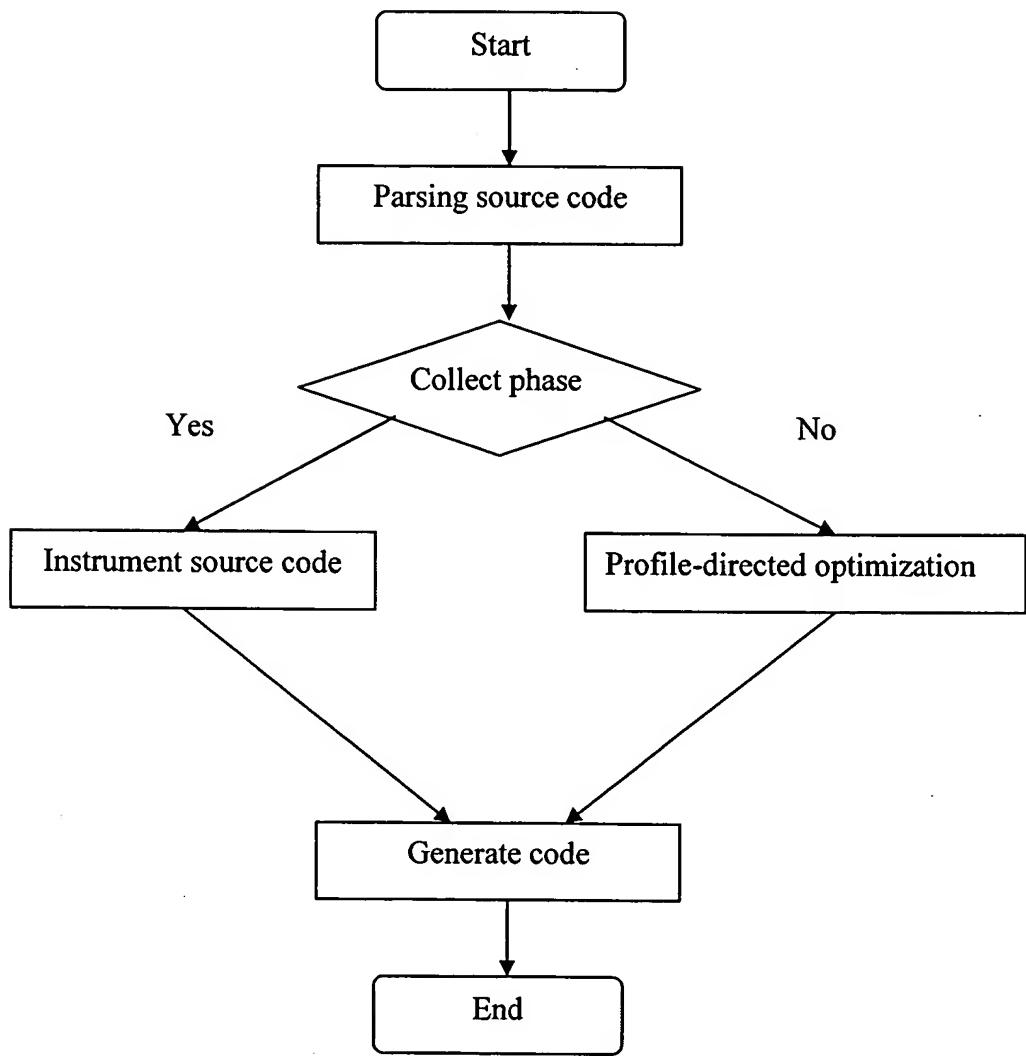


Fig. 5 Frequency change before and after inlining

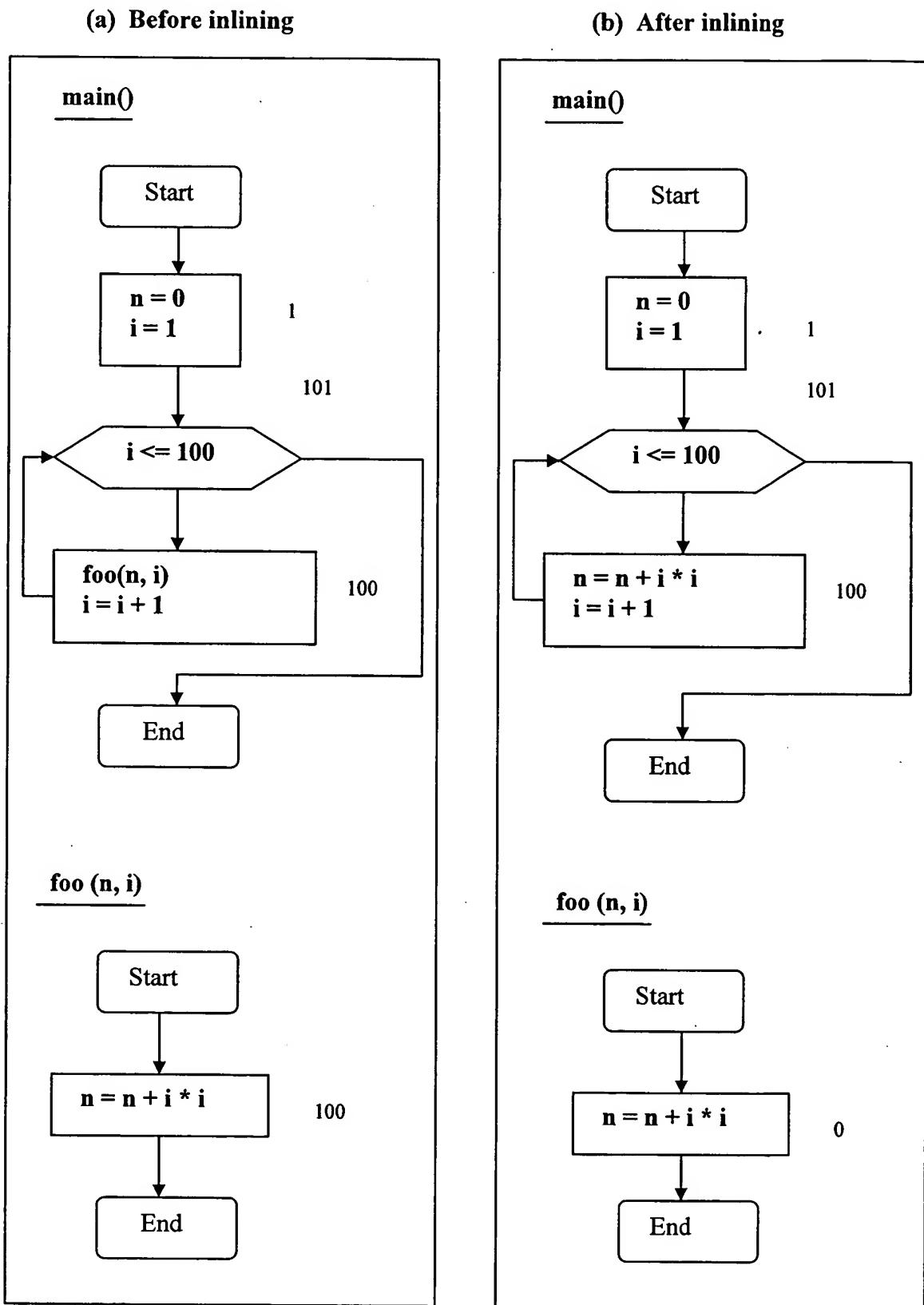


Fig 6. Profile-Directed Optimizations (PDO) (prior art)

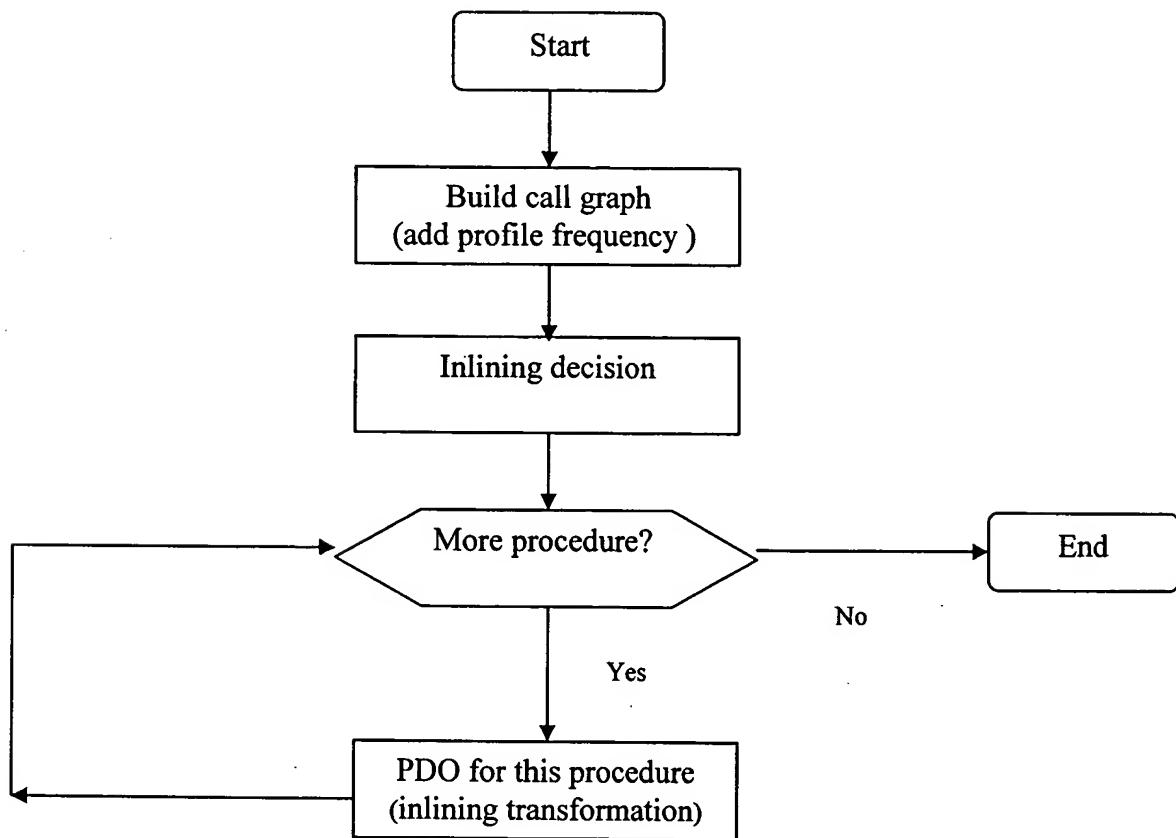


Fig 7. Example of Call graph and IP

Fig 7(a) Original Code

```

foo() {
c1: bar(1);
c2: bar(3);
}

bar(n) {
c3: cat(n);
c4: cat(n+1);
}

cat (m) {
<body of cat(m)>
}

```

Fig 7(b) After Inlining

```

Foo() {
<body of cat(1)>
<body of cat(2)>
c3_2: cat(3)
<body of cat(4)>
}

bar(n) {
cat(n);
cat(n+1);
}

cat (m) {
<body of cat(m)>
}

```

Fig 7(c) Call Graph

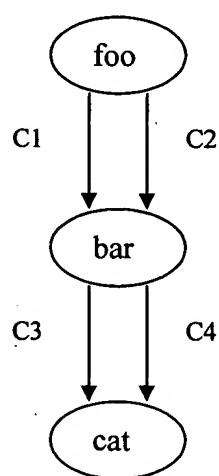


Fig 7(d) IP(foo)

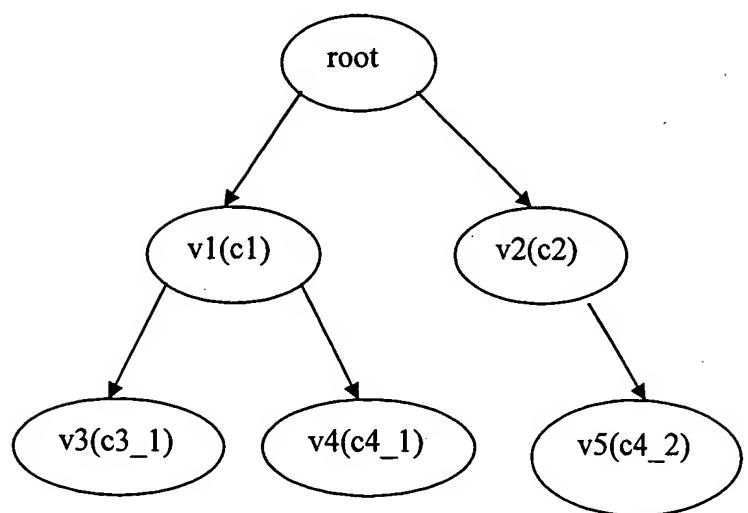


Fig. 8 Inlining Original Procedures

Fig. 8(a) Original Procedures

Fig. 8(a) Original Procedures

```
main () {          foo() {          bar () {  
    foo();          bar();  
}  
}  
}
```

Fig. 8(b) Inlining original bar into foo

```
main () {          foo() {          bar () {  
    foo();          }          }  
}  
}
```

Fig. 8(c) Inlining original foo into main

```
main () {          foo() {          bar () {  
    bar();          }          }  
}  
}
```

FIG. 9 Updating Frequency

```
proc {  
    e;    freq(e)  
}
```

Fig 9(a)

```
callee(e) { freq(e) ↓  
    e1;    a1 ↓  
}
```

Fig 9(b)

```
callee(e1) { a1 ↑  
    e2;    a2 ↑  
}
```

Fig 9(c)

```
Callee(e2) { a2 ↓  
}
```

Fig 9(d)

Fig. 10 Example of Applying the Algorithm

Fig. 10(a) Original procedures (frequency is number after colon)

```
proc() { :1          foo() { :1          bar() { :101          cat() { :101
  e1: foo() :1          for (i=0; i<100;i++)      e4: cat() : 101
  e2: bar() : 1          e3: bar() : 100      }
}                      }
}
```

Fig. 10(b) Inlining e3 into foo

```
proc() { :1          foo() { :1          bar() { :1          cat() { :101
  e1: foo() :1          for (i=0; i<100;i++)      e4: cat() : 1
  e2: bar() : 1          e5: cat() : 100      }
}                      }
}
```

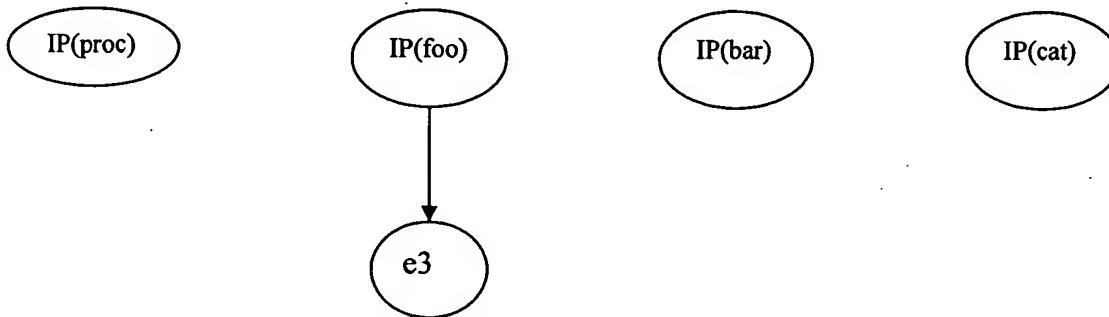


Fig. 10(c) Inlining e4 into bar

```

proc() { :1
  e1: foo() :1
  e2: bar() :1
}
  foo() { :1
    for (i=0; i<100;i++)
      e5: cat() :100
  }
  bar() { :1
  }
  cat() { :100
}

```

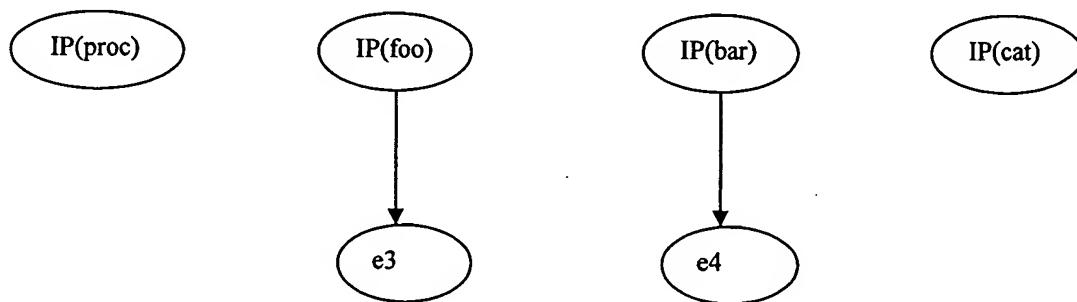


Fig. 10(d) Inlining e1 into proc

```

proc() { :1
  for(I=0;I<100; I++)
    e6: bar() :100
  e2: bar() :1
}
  foo() { :0
    for (i=0; i<100;i++)
      e5: cat() :0
  }
  bar() { :101
}
  cat() { :0
}

```

